AD-A212 858

DTIC ELECTE SEP 27 1989 D

B

REPORT DOCUMENTATION PAGE

1b RESTRICTIVE MARKINGS	
n/a since unclassified	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE n/a since unclassified	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 89-1235
6a NAME OF PERFORMING ORGANIZATION JAYCOR; 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research/NP
6c. ADDRESS (City, State, and ZIP Code) 11011 Torreyana Road San Diego, CA 92121	7b ADDRESS (City, State, and ZIP Code) Building 410 Bolling AFB, DC 20332-6448
8a NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR; 8b OFFICE SYMBOL (If applicable) NP	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER Contract No. F49620-87-C-0062
8c. ADDRESS (City, State, and ZIP Code) Bldg. 410 Bolling AFB, DC 20332-6448	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. 61102F; PROJECT NO 2301; TASK NO A8; WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

NTRFACE FOR MAGIC

12 PERSONAL AUTHOR(S) N.T. GLADD

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Final Report	FROM 870701 TO 890731	890731	38

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	User's manual for NTRFACE Code, NTRFACE is an object oriented interface system for the MAGIC Particle-in-Cell Simulation Code.

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The NTRFACE system was developed under the auspices of the basic physics research program of the Air Force Office of Scientific Research. It's fundamental purpose is to demonstrate the utility of contemporary computing technologies for facilitating the operation of highly complex, but more traditional, scientific computational models. To this end, NTRFACE makes use of the user friendliness of personal computers, the flexibility and expressiveness of modern languages such as LISP and C, the power and extensibility of object-oriented programming and also borrows some techniques from artificial intelligence.

The NTRFACE system is made concrete by applying it to a specific application- a mature, highly complex plasma physics particle in cell simulation code name MAGIC.

This document is intended to be the user manual for the beta test version of NTRFACE for MAGIC. It is not intended to be a finished product! Indeed, the purpose of this release is to distribute NTRFACE for MAGIC to various MAGIC users who will field test it. (continued)

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL DR Robert J. Barker	22b TELEPHONE (Include Area Code) (202) 767-5011	22c OFFICE SYMBOL NP

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

89 9 27 035

BLOCK 19: It is anticipated that bugs, oversights and conceptual errors will be discovered and any comments and constructive criticism will be most welcome.

AFOSR-TR- 89-1235

PREFACE

The NTRFACE system was developed under the auspices of the basic physics research program of the Air Force Office of Scientific Research. It's fundamental purpose is to demonstrate the utility of contemporary computing technologies for facilitating the operation of highly complex, but more traditional, scientific computational models. To this end, NTRFACE makes use of the user friendliness of personal computers, the flexibility and expressiveness of modern languages such as LISP and C, the power and extensibility of object-oriented programming and also borrows some techniques from artificial intelligence.

The NTRFACE system is made concrete by applying it to a specific application- a mature, highly complex plasma physics particle in cell simulation code name MAGIC.

This document is intended to be the user manual for the beta test version of NTRFACE for MAGIC. It is not intended to be a finished product! Indeed, the purpose of this release is to distribute NTRFACE for MAGIC to various MAGIC users who will field test it. It is anticipated that bugs, oversights and conceptual errors will be discovered and any comments and constructive criticism will be most welcome.

TABLE OF CONTENTS

1. INTRODUCTION

This manual describes NTRFACE for MAGIC, an object-oriented interface system for the MAGIC Particle-In-Cell simulation code. Although much of the following discussion will be specific to the MAGIC code, the NTRFACE [1] system itself was developed as part of a general research program aimed at utilizing contemporary computational tools in order to facilitate the use of complex scientific computer models. Many of the features of the NTRFACE system are generic to problems of scientific computing. Since the underlying architecture is based on the flexible object-oriented programing paradigm, NTRFACE, with modest redesign, may be adapted to provide interfaces for other scientific computational models.

Particle-In-Cell plasma simulation codes[2], often called PIC codes, are widely used in plasma physics research and are one of the most important tools for advancing plasma science. PIC codes are sophisticated computational models, developed and almost exclusively used by plasma physicists. To say the least, PIC codes are difficult to use. However, the power and flexibility of PIC codes makes it desirable that they be used in areas in addition to plasma research. For example, PIC codes could be very useful for electronics engineers involved in microwave tube design. It is hoped that the NTRFACE system will make PIC codes more accessible to nonexperts, increase the productivity of working plasma physicists and, perhaps, decrease the time graduate students require to become proficient with PIC codes.

MAGIC [3,4] is a two-dimensional, finite-difference time-domain code for simulating plasma physics processes involving interactions between space charge and electromagnetic fields. It is an example of a particle-in-cell or PIC code and is specifically focused on treating complicated boundary and/or initial conditions. It's name is an acronym deriving from MAGnetic Insulation Code. MAGIC is a mature plasma simulation code which can be used in a variety of applications- basic plasma physics, pulsed power, charged-particle beams, vacuum electronics and others. MAGIC is currently being be distributed to Universities where it will be put to a variety of other uses.

MAGIC is also a challenging test for the NTRFACE system. The difficulty of interfacing with MAGIC is exemplified by the following quote from it's authors [4]- "*Although the MAGIC code is designed to be user friendly, our experience has been that a new user will typically require several years to be proficient in it's use.*" These authors explain that this difficulty of access results because use of these codes requires knowledge of the underlying mathematical equations, knowledge of the physical situations being modeled and, in addition, practical knowledge about the tractability of these physical situations to numerical solution. We can add to this list. PIC simulations often involve complex interactions between simulated objects whose comprehension requires visualization. Input parameters defining simulations are constrained in complex ways, often depending critically on each other's values. Furthermore, the rigid syntax of FORTRAN, the standard language for PIC simulations, requires input to be specified in terse sets of numbers. Such cryptic formatting hides physical intuition and is also susceptible to typographical or transcription errors.

Because of the complexity of input required for MAGIC, it's authors have written a metalanguage which facilitates input by taking some advantage of the structure of the input. The syntax of an input statement to MAGIC has the form

<ALPHANUMERIC STRING> <PARAM1> <PARAM2> <PARAM3>... /

Consider the specific example

CONDUCTOR CATHODE1 ALIGN 11 10 51 10 /

Here *CONDUCTOR* is an alphanumeric string instructing MAGIC's input parser to read the remaining parameters as would be expected to format a conductor. *CATHODE* is an alphanumeric string parameter naming a particular conducting surface; ALIGN is an another string parameter specifying the normal of the conducting surface; the four integers specify the starting (i_{xs}, i_{ys}) and ending (i_{xe}, i_{ye}) grid points of the conducting surface; the symbol "/" is a token signifying termination of input. There are many MAGIC commands and the input lines can be much more complicated than this simple example. The MAGIC input language is described in detail in reference [3].

The NTRFACE for MAGIC system operates by facilitating the setup of input files for the MAGIC code. It is not intrusive on MAGIC. Indeed, the NTRFACE system does not reside on the same computer as MAGIC. NTRFACE for MAGIC leads users through a series of visualizations, menus and data entry forms which enable them to parameterize a MAGIC simulation. At the end of a session, NTRFACE for MAGIC writes a disk file containing lines of input which can be directly interpreted by MAGIC. There are a number of advantages in using NTRFACE to set up a MAGIC simulation:

1) Basic MAGIC objects, such as the computational grid, conductors and boundary conditions can be viewed as they are parameterized. Incremental changes can be quickly made by moving objects with a mouse.

2) The likelihood of erroneous input is greatly reduced. Some objects are constrained so that they can only assume legal positions and orientations. Other parameters must satisfy rules which restrict them to acceptable values.

3) MAGIC input lines are automatically written by NTRFACE for MAGIC. This avoids transcription and/or typographical errors. Also, a user does not need any knowledge of formatting requirements.

4) Once a MAGIC case has been set up, it is stored in a library as a collection of objects, as well as an input file for MAGIC, and can be used as a template for setting up subsequent MAGIC cases.

2. THE FUNCTION OF NTRFACE FOR MAGIC

The specific function of NTRFACE for MAGIC is to create input files for the MAGIC simulation code. It does not "run" the MAGIC code. Indeed, MAGIC is typically resident on a main frame computer while NTRFACE for MAGIC resides on a personal computer. Thus, an input file created with NTRFACE for MAGIC must, in some fashion, be transferred to a computer capable of running MAGIC.

NTRFACE for MAGIC allows users to visualize, manipulate and parametrize the various objects which constitute a run of the MAGIC simulation code. Such objects include the two-dimensional computational grid, conducting surfaces, boundary conditions, emission models, mathematical functions defining temporal and spatial variations, algorithms and diagnostics. Other objects are sets of parameters which produce various forms of output.

Once a user has constructed a set of simulation objects for a MAGIC run, the NTRFACE for MAGIC system automatically specifies the input lines for these objects using the syntax of the MAGIC input language. This is best illustrated by an example (see Figure 1).

Unfortunately, the various color screens which appear in NTRFACE for MAGIC had to be reproduced in monochrome within this manual. To see the Figures as they actually appear place the NTRFACE for MAGIC disk in drive A: of an IBM PC or compatible with EGA color graphics and 1) initialize the mouse driver, 2) type A:\FIGURES.

In Figure 1, an NTRFACE for MAGIC screen is depicted on which the computational grid and several conducting surfaces are visible. At the top of the screen is a menu of operations which are relevant to conducting surfaces. A surface may be ADDed, DELeted, MODified, or DEScribed. The remaining option is CONtinue which returns control to the Main Control Menu. A user mouses on one of these menu items (moves the cursor into red box containing item and clicks left key) and follows subsequent instructions to place, move, reshape or examine a conducting surface. Later, after all other simulation objects have been specified, the conducting surface objects displayed in this figure would be translated into the ASCII character strings

```
CONDUCTOR CON1 ALIGN 2 12 2 12 /
CONDUCTOR CON2 ANTI-ALIGN 12 2 12 9 /
CONDUCTOR CON3 ALIGN 12 9 17 14 /
CONDUCTOR CON4 ALIGN 17 14 42 14 /
CONDUCTOR E1 ANTI-ALIGN 2 22 52 22 /
```

which will be part of the input file for MAGIC.

Several things are notable. It is easier and more natural to design and modify a set of conductors by manipulating the screen than by editing the ASCII strings. Additionally, the NTRFACE for MAGIC system constrains conducting surfaces to the computational

grid, and doesn't permit an illegal orientation (whereas a typographical or transcription error could result in a configuration which would cause MAGIC to fail.) Users do not have to remember the syntax of the MAGIC input language. The syntax for conductors is straightforward- the name of the surface, the alignment of the surface normal, and the grid points of the surface end points. However, other simulation objects require input lines with many numerical and alpha-numeric parameters specified in precise order.

While NTRFACE for MAGIC reduces the ways in which erroneous input can occur, *they are not eliminated*. For example, within MAGIC the combination of conductors, emitters and boundaries must form a composite surface enclosing the simulation region. This is not currently checked by NTRFACE for MAGIC. Such a gap, however, would be glaringly apparent on the screens which display the simulation objects. More subtle errors can occur. The placement of two emitting surface may not allow sufficient spatial resolution (grid points) to accurately describe the physics of interest. While such types of errors are not currently handled by NTRFACE for MAGIC, the underlying object-oriented approach is conducive to the development of procedures for treating such errors. Standard rule-based expert systems could be used to check for such problems. A basic requirement for such an approach is that rules relating objects can be appropriately articulated.

An important function of NTRFACE for MAGIC is the ability to store, in a reusable format, the collection of objects, called a "case", which constitutes a MAGIC simulation. This permits incremental modification of existing cases. In fact, users of NTRFACE for MAGIC always work with an existing case- either one of a library of example cases, or a previously created case.

3. HOW TO USE NTRFACE FOR MAGIC

SYSTEM REQUIREMENTS:

NTRFACE for MAGIC requires an IBM PC, PC-AT or 100% compatible computer with a hard disk, EGA color graphics, and a mouse. The system was developed on DELL System 325 and 310 (80386 based) microcomputers.

INSTALLATION:

To install NTRFACE for MAGIC

1) Make sure your mouse driver has been installed.

2) Place the NTRFACE for MAGIC disk in drive A: and type INSTALL.

The installation procedure will create two directories, NTRFACE and NTRDATA on the C: hard drive. The NTRFACE for MAGIC system and supporting graphics are in the NTRFACE directory. The NTRDATA directory contains a library of example cases. Any cases subsequently created by users will also reside in this library. Each case has three associated files with suffixes .CAS (the reusable simulation objects), .OUT (the reusable output objects), and .DCK (the input deck for MAGIC). The *.DCK files are ASCII files and may, if desired, be edited with any text editor. The *.CAS and *.OUT are binary representations of C language storage structures and may be conveniently changed only by the NTRFACE for MAGIC system. Any case which is in the NTRDATA directory may be modified by NTRFACE. If cases need to be preserved for reference purposes, they should be copied to another directory.

STARTING THE PROGRAM:

To start the NTRFACE for MAGIC program, type

1) cd \NTRFACE *(change directories)*

2) NTRFACE *(start the program)*

An introductory screen will appear with instructions for proceeding.

OVERVIEW OF PROGRAM OPERATION:

The basic procedure during an NTRFACE for MAGIC session consists of the following.

1) Choose a case to modify. A case is a collection of simulation objects which parameterizes a MAGIC simulation and it's

various output forms. A case is chosen from the library of preexisting cases. If in doubt, select the DEFAULT case.

2) Modify the case as desired. This is accomplished by cycling through various menus and data entry forms. Movement through these menus is controlled by two master control menus- the Main Control Menu which controls MAGIC features associated with the simulation model, and the Output Control Menu, which controls the various options for output.

3) Save the case to the disk library. Saving a case produces an input deck for the MAGIC code and also adds the saved case to the library of cases available for subsequent modification. A case does not have to be saved. The program can be terminated without saving a case at any time by selecting the ABORT option on the Main Control Menu.

As a prelude to discussion of the detailed features of NTRFACE for MAGIC, the basic menuing and data entry system are described.

HOW TO USE MENUS AND DATA ENTRY FORMS:

There are two types of menus within NTRFACE for MAGIC- operation menus and choice menus. For both menu types all selections are made with the left mouse key.

<u>OPERATION MENUS</u> are for choosing one of several possible operations (see Figure 1).

A frequently occurring operation menu within NTRFACE for MAGIC appears across the top of the screen in Figure 1 and presents the operations CONtinue, ADD, DELete, MODify and DEScribe. An operation is selected by moving the mouse cursor within one of the red boxes and clicking the left most key. The latter four operations refer to standard operations that can be performed on various simulation objects. For example, one can add, delete, describe or modify an "emitter object" just as one can add, delete, modify or describe a "contour plot object." The operation CONtinue simply continues to the next menu or data entry form. The Main Control Menu (Figure 5) is another example of a choice menu. If the lettering of one of the operations is yellow, it is the default operation.

<u>CHOICE MENUS</u> are for choosing the alphabetic or alphanumeric value of a parameter from a list of possible values (see Figure 2).

In this choice menus, users are choosing a direction for the normal to a conducting surface. The available choices are ALIGN and ANTI-ALIGN, i.e., with or against the direction of the coordinate axis. A choice is made by mousing on the red rectangle containing the desired value. The default value appears in yellow letters. Default, in this respect, means that this is the current value of that attribute for the underlying object which

is being modified. It is important to realize that NTRFACE for MAGIC always acts to modifying an existing object. Even if a new object (say conducting surface) has just been ADDed, it is first assigned default values which may then be modified. Notice also the green rectangles with the work HELP to the right of each choice. These rectangles are mouse active and selecting one of them will result in a pop up screen containing information about it's corresponding item.

DATA ENTRY FORMS are for assigning the value of numerical parameters (see Figure 3).

This data entry form depicts some of the parameters characterizing a perspective plot. The name of the parameter appears at left and it's current value is in a red box (if integer) or brown box (if float). If a new value for one of the parameters is desired, the appropriate red or brown rectangle is moused and the a numerical value is entered by typing a number followed by the ENTER key. The entered value must be a number; illegal sequences of characters will not be accepted (an error message will appear and a legal number must be entered.) If an integer is entered for a float, it is converted. If a float is entered for an integer, it is truncated. When all changes have been made, the green rectangle denoted CONTINUE is moused. As with choice menus, each of the parameters has a HELP box associated with it.

There is also a limited error trapping capability within NTRFACE for MAGIC. Each numerical parameter has an acceptable minimum and maximum value attached to it and entering a value outside this range will result in an error message. In this version of NTRFACE for MAGIC the limiting values have been estimated. It is quite likely that they will have to be modified as user experience develops.

MANIPULATING THE DISPLAY SCREEN:

At various points within NTRFACE for MAGIC, a screen is displayed which shows part or all of the computational grid, as well as surfaces such as conductors, emitters and boundaries. This provides a guide for the modification of the composite system.

In Figure 5, the screen shows a 50x20 portion of a 70x20 Cartesian grid with conducting surfaces and a boundary surface. The numbers at the ends of the abscissa and ordinate show the limiting grid points and corresponding physical values of those grid points. Also shown are some two grey rectangular regions partially filled with green. The relative amount of green filling indicates the portion of the complete grid which is currently visible (abscissa and ordinate). In the upper left hand corner, a green box containing the word VIEW is mouse active. Clicking on this box produces a data entry form which allows the parameters of the viewing window to be specified. The parameters which may be set are the grid points of the lower left hand corner and the width and height (in grid points) of the visible window. This example screen is part of the process of parameterizing boundaries and the operation menu at the top of the screen pertains to the boundary surfaces.

CENTRAL CONTROL:

The NTRFACE for MAGIC system permits users to specify a number of the features of the MAGIC code. For organizational purposes, these features are categorized as SIMULATION features (parameterizing the simulation being prepared for the MAGIC code) and OUTPUT features (parameterizing the various formats for MAGIC output.) Individual SIMULATION and OUTPUT features will be described in the next section.

The Main Control Menu, an operation menu, permits selection of particular SIMULATION features (see Figure 5).

Mousing on a red box passes control to the menus and data entry forms which parameterize that SIMULATION feature. In some instances, a visual guide is presented to assist the specification of parameters. For example, when BOUNDARIES is selected, a screen displaying the computational grid, the existing conducting and emitting surfaces and the existing boundaries forms the background (see Figure 4). After parameters corresponding to a particular feature have been specified control is returned to the Main Control Menu and the next feature may be selected in turn. Although a sequence for parameterizing features is suggested by the Main Control Menu, any particular feature may be specified at any time. The item marked OUTPUT transfers control to the Main Output Control Menu (see Figure 6).

The Output Control Menu, an operation menu, permits various OUTPUT features to be parameterized. The procedure is to select the desired OUTPUT feature and fill in the various menus and data entry forms parameterizing that type of output. Since multiple versions of most of the MAGIC OUTPUT features are possible, a list of the already existing versions of that OUTPUT feature is shown and users can either ADD to this list or DELete, MODify, or DEScribe any existing item.

SAVING A CASE:

After all SIMULATION and OUTPUT features have been parameterized, SAVE CASE is selected from the Main Control Menu. Users then select a name for the new case (selecting the default name will overwrite the existing case). Finally, users may also add up to three lines of comments describing this case. These comments will be incorporated in the MAGIC input deck by means of the MAGIC TITLE feature.

Before terminating, NTRFACE for MAGIC will write three output files to the NTRDATA directory. If the case name was chosen to be ABC, then the three output files are ABC.CAS (binary representation of the SIMULATION objects), ABC.OUT (binary representation of the OUTPUT objects) and ABC.DCK (ASCII representation of the MAGIC input lines). Upon restarting the NTRFACE for MAGIC system, the case ABC will be found in the library of cases and available for modification.

4. COMPONENTS OF NTRFACE FOR MAGIC

OVERVIEW:

The MAGIC code has evolved over a number of years and contains a large number of features. This version of NTRFACE for MAGIC does not attempt to encompass all of the capabilities of MAGIC. The basic functions that NTRFACE for MAGIC currently included are-

* choice of geometry
* specification of computational grid
* placement of conducting elements within the grid
* assignment of emission characteristics to conducting elements
* placement and parameterization of boundary conditions
* specification of computational algorithms
* specification of a number of kinds of output forms

NTRFACE for MAGIC can presently be used to set up and modify the following MAGIC features:

BEAM EMISSION	BEXTERNAL	COMPLOT
CONDUCTOR	CONTOUR	COURANT
EMIT	FIELD EMISSION	FIELDS
FORCES	FUNCTION	KINEMATICS
LINPRINT	LOGPRINT	LOOKBACK
OBSERVE	OUTPUT	PARTICLES
RANGE	START	STATISTICS
SYMMETRY	SYSTEM	TAGGING
TITLE	VECTOR	VOLTAGE
X1GRID	X2GRID	

These features of MAGIC will now be described in the order they appear on the CONTROL menus. A detailed description of each of these MAGIC features may be found in MAGIC USER'S MANUAL[3]. References in the following section refer to page numbers in this manual.

The extension of NTRFACE for MAGIC to include additional MAGIC features is straightforward and more will be implemented in subsequent versions.

GEOMETRY

Corresponding MAGIC command: SYSTEM

Reference: 5-137

Function: Specifies the coordinate system.

Description: CHOICE MENU to select one of four available coordinate systems.

GRIDS

Corresponding MAGIC command: X1GRID, X2GRID

Reference: 5-154

Function: Generates spatial grids in the x and y directions.

Description: The existing computational grid is depicted on the screen. Specifically, the present geometry is indicated at the lower left and the coordinates (grid point and physical value) are indicated on the respective axes.

Three menu choices are available: 1) CONtinue, 2) MODify XGRID(abscissa), 3) MODify YGRID(ordinate). Choosing to MODify a grid causes a data entry form to appear. The grid points and physical values of the ends of an axis may be specified. The final item, STRETCH, permits a quadratically nonuniform grid to be specified (see 5-155). The key parameter determining the grid spacing is the ratio

STRETCH = DX/DXuni

where is DX is the desired spacing of the first two grid points and DXuni is the spacing if the interval is divided uniformly. Thus

0.01 <= STRETCH < 1.0	spacing increases with coordinate
STRETCH = 1.0	uniform spacing with coordinate
1.00 < STRETCH <= 1.98	spacing decreases with coordinate

Each axis may be independently specified.

The NTRFACE for MAGIC system currently permits only one grid to be parameterized. Within MAGIC, multiple grids may be specified.

CONDUCTORS

Corresponding MAGIC command: CONDUCTOR

Reference: 5-24

Function: Specifies perfectly conducting surfaces in the spatial grid.

Description: A screen with the grid and any existing conducting surfaces (actually line segments) will appear. A new conducting surface can be ADDed, or an existing surface can be DELeted, MODified, or DEScribed. If the choice ADD is made, a red box will appear which prompts the user to select the left hand or upper grid point (as viewing the screen). This is accomplished by moving the cursor to the desired grid point and clicking the left mouse button. A yellow circle will denote the choice. The same procedure is followed for the other end of the surface. Next, the orientation of the normal to the conducting surface is specified, either ALIGNED or ANTI-ALIGNED with one of the axes. Finally, a name must be specified for the conducting surface. These names are important because emission models are associated with a given conducting surface by means of it's name. When an operation is finished on a conducting surface, the screen will refresh and depict any changes which have been made. Conducting surfaces must be parallel to, or exactly diagonal to the grid axes. Error traps prevent incorrectly oriented conductors from being placed on the grid.

If the other operations are selected, a message will appear requiring the selection of an existing conducting surface. This is accomplished by mousing within the desired object. Only a circular region near the center of the conductor is mouse active so it may take more than one mouse click to select a conductor.

Within MAGIC, emitting surfaces are created by attaching emission models to conducting surfaces. Thus, emitting surfaces should first be specified as conducting surfaces and later modified to be emitters.

EMISSION MODELS

Corresponding MAGIC command: BEAM-EMISSION, FIELD-EMISSION

Reference: 5-9, 5-54

Function: Specifies beam and field emission models.

Description: A box describing the existing beam emission models will appear. This list may be ADDed to or any model may be DELeted, MODified, or DEScribed. An object is selected by mousing anywhere within it's box. There are a number of parameters which specify an emission model and a sequence of menus and data entry forms are provided to enable the parameters to be specified. In addition to specifying parameters, a name for the emission model must be chosen. This name is important because emission models are attached to conducting surfaces by means of this name. The name is also used to construct unique names for the function describing spatial and temporal features of the emission models.

Both beam and field emission models are specified before returning to the Main Control Menu.

EMITTERS

Corresponding MAGIC command: EMIT

Reference: 5-46

Function: Enables emission on specified surface.

Description: A screen appears with the grid and all existing conducting and emitting surfaces depicted. Various color coding and fill patterns distinguish the various types of surfaces. These can be identified by DEScribing any desired object in the CONDUCTORS section. Menu options are to CONTINUE (return to Main Control Menu) or MAKE EMIT (transform a conductor into an emitter). On choosing MAKE EMIT, a conducting surface must first be selected. Next, choice of beam or field emission is made. Finally, a selection from the list of existing beam or field emission models is made.

In specifying emitters, it may be necessary to move back and forth between the CONDUCTORS, EMISSION MODELS and EMITTERS sections. Conductor geometry and orientation are specified with CONDUCTORS, emission model parameters are specified with EMISSION MODELS. The EMITTER section allows a specific emission model to be attached to a specific conducting surface.

BOUNDARIES

Corresponding MAGIC command: LOOKBACK, SYMMETRY, VOLTAGE

Reference: 5-81, 5-135, 5-150

Function: LOOKBACK provides an outlet for an electromagnetic wave to escape from the simulation. SYMMETRY specifies various, symmetry based boundary conditions. VOLTAGE specifies an incident electromagnetic wave which will enter the two-dimensional simulation region.

Description: A screen appears depicting the grid, existing conducting and emitting surfaces and any existing boundary surfaces. A new boundary can be ADDed or an existing surface can be DELeted, MODified, or DEScribed. There are five different kinds of boundary conditions currently implemented within NTRFACE for MAGIC- LOOKBACK, VOLTAGE, AXIAL, MIRROR and PERIODIC. The latter three are implemented with MAGIC's SYMMETRY command. The procedure for adding a new boundary surface is identical to that of adding a conducting surface. Some of the boundary condition require the specification of various parameters by means of menus and/or data entry forms.

BEXTERNAL

Corresponding MAGIC command: BEXTERNAL

Reference: 5-13

Function: Specifies spatially independent static magnetic fields.

Description: A data entry form appears which enabled the specification of the magnitudes of the magnetic field components.

FIELDS

Corresponding MAGIC command: FIELDS

Reference: 5-59

Function: Specifies electromagnetic field algorithm and parameters.

Description: A sequence of menus and data entry forms for the various parameters.

KINEMATICS

Corresponding MAGIC command: KINEMATICS

Reference: 5-75

Function: Specifies particle kinematics algorithm.

Description: A sequence of menus and data entry forms for the various parameters.

FORCES

Corresponding MAGIC command: FORCES

Reference: 5-65

Function: Specifies the spatial averaging parameters used to calculate electromagnetic forces for particle kinematics.

Description: A sequence of menus and data entry forms for the various parameters.

CURRENTS

Corresponding MAGIC command: CURRENTS

Reference: 5-32

Function: Specifies current density algorithm and parameters.

Description: A sequence of menus and data entry forms for the various parameters.

COURANT

Corresponding MAGIC command: COURANT

Reference: 5-30

Function: Checks for COURANT stability condition.

Description: Menu which turns on global COURANT condition check. Capability for checking a specific grid point is not currently implemented within NTRFACE for MAGIC.

OUTPUT

This operation passes control to the Output Control Menu where various MAGIC output features can be invoked. Many of the MAGIC output features can be repeated. For example, a number of different contour plots of different field components and different regions can be specified for a given run of the MAGIC code. When a type of output format is requested within the NTRFACE for MAGIC system, a list of all of the existing types of that output format are shown and a user may ADD to that list, or DELete, MODify, or DEScribe any item on that list. Various visual aids will be provided for choosing the spatial regions over which the various output formats are to be applied. Only a certain number of each output form is permitted and a user will be warned if that limit is about to be exceeded.

After all output features have been selected the user will be prompted to choose the medium for output. The choices are SYSTEM (use the graphics capabilities of the computer system on which MAGIC is implemented) or PRINTER (line printer output). This choices sets the OUTPUT command of MAGIC(5-96).

LIN/LOG PRINTS

Corresponding MAGIC command: LINPRINT, LOGPRINT

Reference: 5-77, 5-79

Function: LINPRINT specifies printing a field component as a table of numbers, LOGPRINT specifies printing a field component as an array of characters representing ranges of values.

Description: Menus and data entry forms are presented for selecting the field component and specifying other parameters. Particularly important parameters are the initial, incremental and final times for which output is desired. A screen is displayed which depicts the region (dashed yellow lines) over which the component will be printed. This rectangular region is modified by choosing a new upper left and lower right point.

CONTOUR PLOTS

Corresponding MAGIC command: CONTOUR

Reference: 5-26

Function: Specifies contour plot of field components.

Description: Menus and data entry forms are presented for selecting the field component and specifying other parameters. Particularly important parameters are the initial, incremental and final times for which output is desired. A screen is displayed which depicts the region (dashed yellow lines) over which the component will be plotted. This rectangular region is modified by choosing a new upper left and lower right point.

COMPLOTS (PERSPECTIVE)

Corresponding MAGIC command: COMPLOT

Reference: 5-19

Function: Specifies perspective plots of field components.

Description: Menus and data entry forms are presented for selecting the field component and specifying other parameters. Particularly important parameters are the initial, incremental and final times for which output is desired. A screen is displayed which depicts the region (dashed yellow lines) over which the component will be plotted. This rectangular region is modified by choosing a new upper left and lower right point.

VECTOR PLOT

Corresponding MAGIC command: VECTOR

Reference: 5-146

Function: Specifies vector plots of two-dimensional fields.

Description: Menus and data entry forms are presented for selecting the two field components and specifying other parameters. Particularly important parameters are the initial, incremental and final times for which output is desired. A screen is displayed which depicts the region (dashed yellow lines) over which the component will be plotted. This rectangular region is modified by choosing a new upper left and lower right point.

RANGE PLOT

Corresponding MAGIC command: RANGE

Reference: 5-119

Function: Specifies simulation variable to be plotted as a function of space along one of the coordinate axes.

Description: Menus and data entry forms are presented for selecting the field component and specifying other parameters. Particularly important parameters are the initial, incremental and final times for which output is desired. A screen is displayed which depicts the line segment (dashed blue line) over which the component will be plotted. This line segment is modified by choosing new end points.

OBSERVE FIELD HISTORY

Corresponding MAGIC command: OBSERVE (TWOD)

Reference: 5-92

Function: Specifies simulation variable to be plotted as a function of time.

Description: Menus and data entry forms are presented for selecting the field component and specifying other parameters. Particularly important parameters are the initial, incremental and final times for which output is desired. A screen is displayed which depicts the line segment (dashed blue line) over which the component will be averaged or the point (blue circle) at which the component is evaluated. The line segment averaged observation is modified by choosing new end points.

OBSERVE ENERGY HISTORY

Corresponding MAGIC command: OBSERVE (ENERGY)

Reference: 5-92

Function: Specifies simulation variable to be plotted as a function of time.

Description: Menus and data entry forms are presented for selecting the type of energy and specifying other parameters. Particularly important parameters are the initial, incremental and final times for which output is desired.

PARTICLE STATISTICS

Corresponding MAGIC command: STATISTICS

Reference: 5-131

Function: Specifies times at which particle statistics are collected and printed.

Description: Menus and data entry forms are presented for selecting parameters.

PARTICLE TAGGING

Corresponding MAGIC command: TAGGING

Reference: 5-138

Function: Specifies which particle are plotted on output graphics.

Description: Menus and data entry forms are presented for selecting parameters.

ACKNOWLEDGEMENTS

I would like to acknowledge Nick Krall, who laid the groundwork for this project, and who always encouraged me to pursue both of my interests, theoretical plasma physics and advanced computation. I would like to thank Bob Barker, who has offered a number of suggestions which have given focus to this project, and who has steadfastly maintained his support in spite of some unexpected twists and turns. I thank Bruce Goplen for answering my questions about MAGIC. Thanks are also due to Dave Sargis for smoothing troubled administrative waters, and to Steve Brecht and Joe Workman for understanding my desire to carry this project to completion.

This research was supported by the Air Force Office of Scientific Research under Contract F49620-87-C-0062.

REFERENCES

[1] N. T. Gladd, "Object-Oriented Interface System for Particle-In-Cell Simulations," to be published, Computational Methods for Vacuum Electronics, special issue of IEEE Transactions on Electron Devices II, 1988.

[2] C. K. Birdsall and A. B. Langdon, Plasma Physics Via Computer Simulation, McGraw-Hill, 1985.

[3] B. Goplen, L. Ludeking, J. McDonald, G. Warren, and R. Worl, "MAGIC USERS MANUAL (September 1988)" Mission Research Corporation Report, MRC/WDC-R-184, Mission Research Corporation, 8560 Cinderbed Road, Suite 700, Newington, VA 22122.

[4] B. Goplen and R. Worl, "Pulsed Power Simulation Problems in MAGIC", Mission Research Corporation Report MRC/WDC-R-124, February, 1987.

[5] N. T. Gladd, "Symbolic Manipulation Techniques for Plasma Kinetic Theory Derivations," J. Computational Physics **56**, 175, 1984.

[6] N. T. Gladd and N. A. Krall, "Artificial Intelligence Methods for Facilitating Large-Scale Scientific Computations," in Coupling Symboic and Numerical Computing in Expert Systems, J. S. Kowalik (editor), Elsevier Science Publishers B. V. (North Holland), 1986.

[7] N. T. Gladd, "ANLYZ: A Symbolic Computing System For Facilitating Scientific and Engineering Computations," JAYCOR Report J530-87-720, 1987.

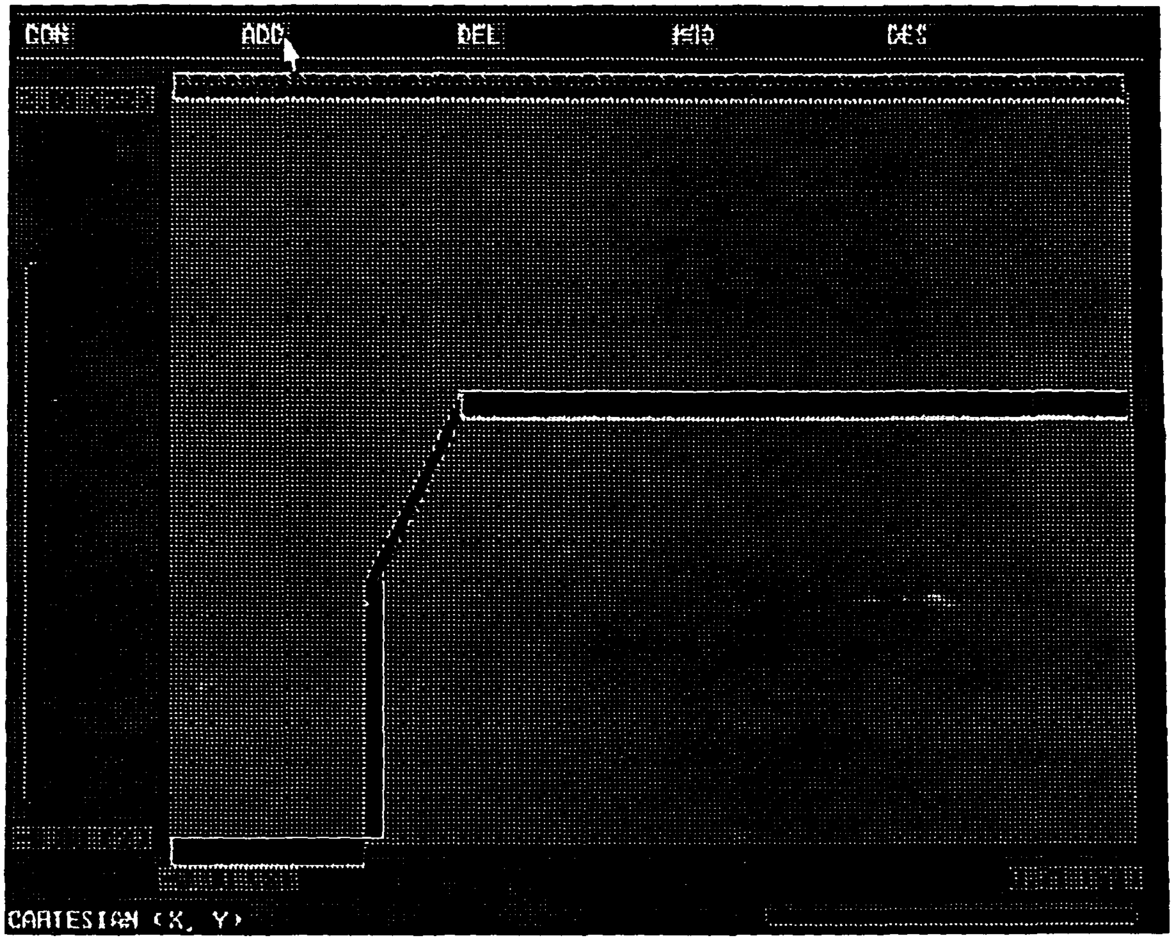

Figure 1 Screen showing conducting surfaces placed on computation grid.

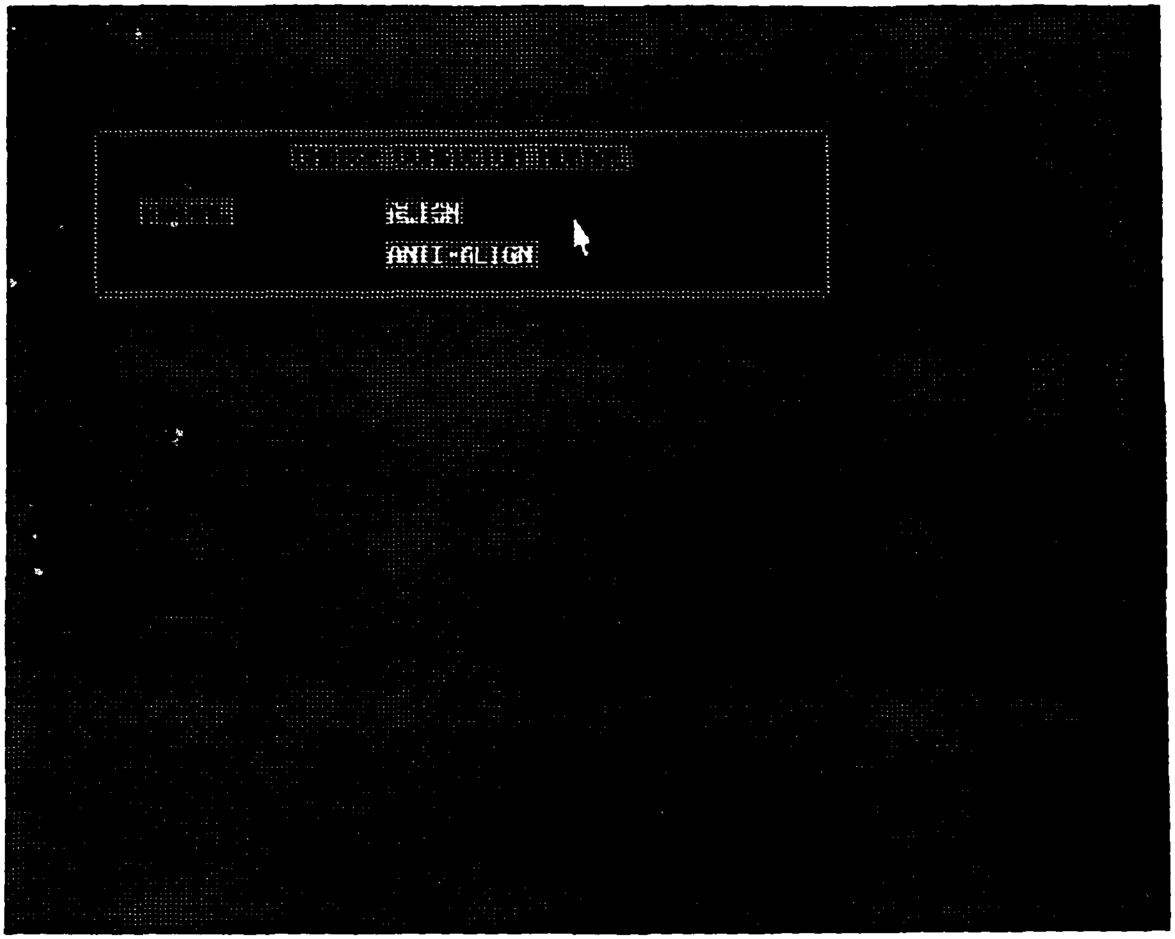

Figure 2 Example of choice menu.

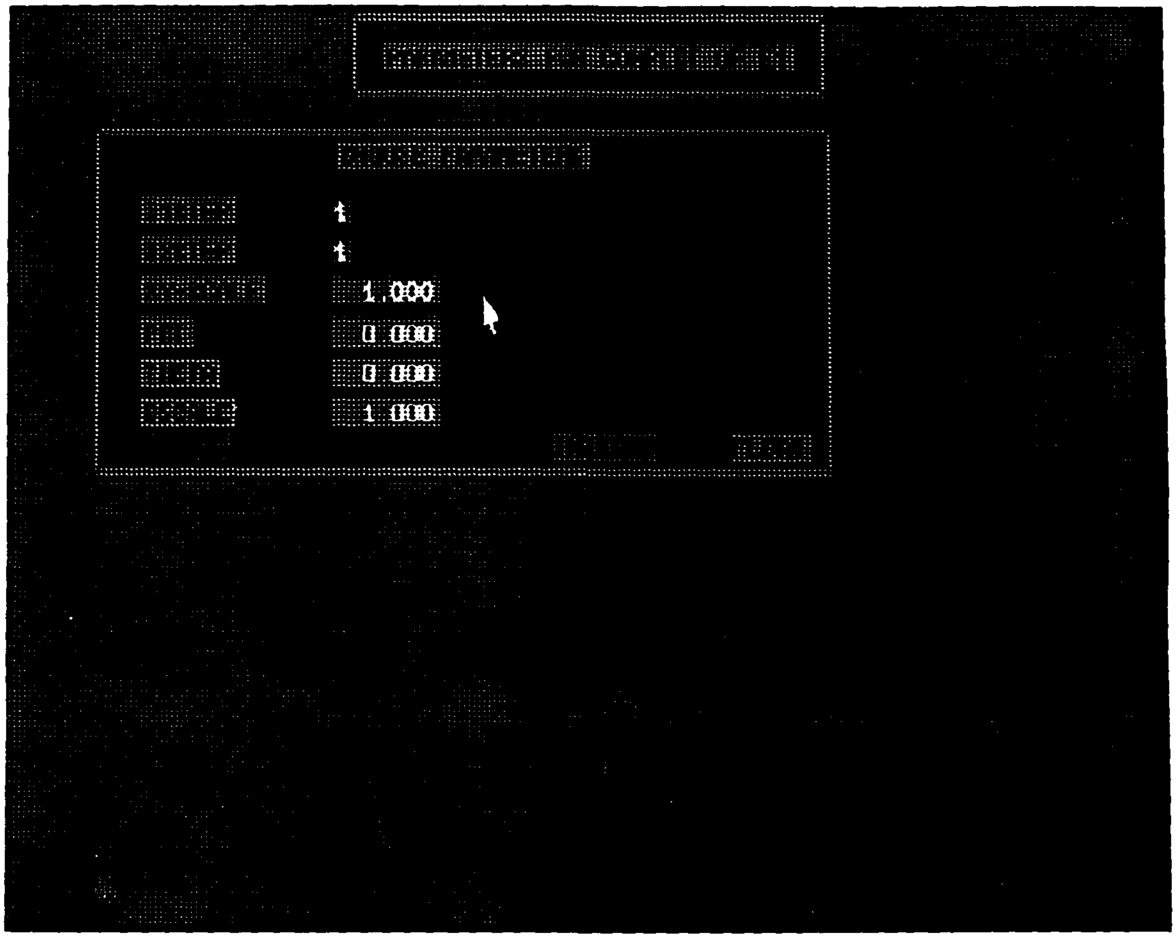

Figure 3 Example of data entry form.

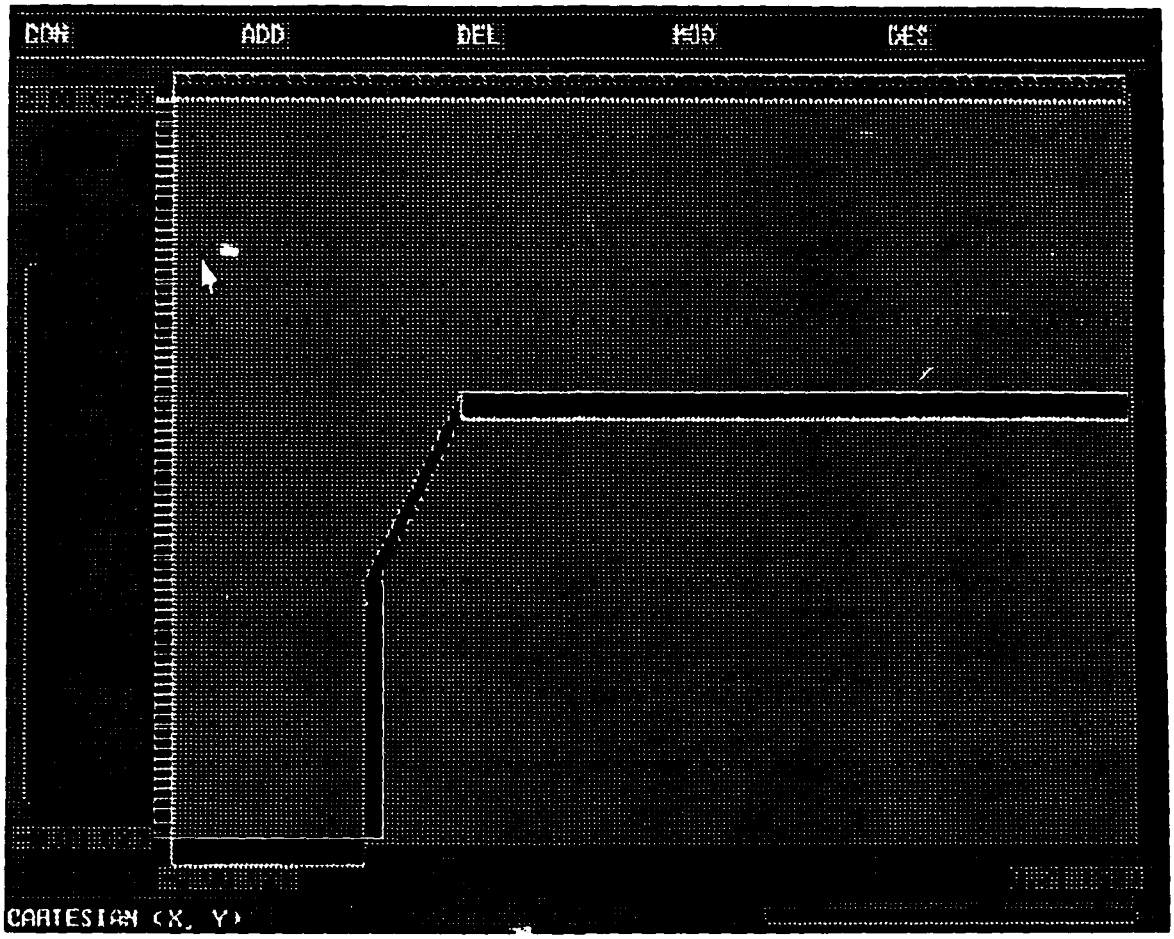

Figure 4 Manipulating the viewing window.

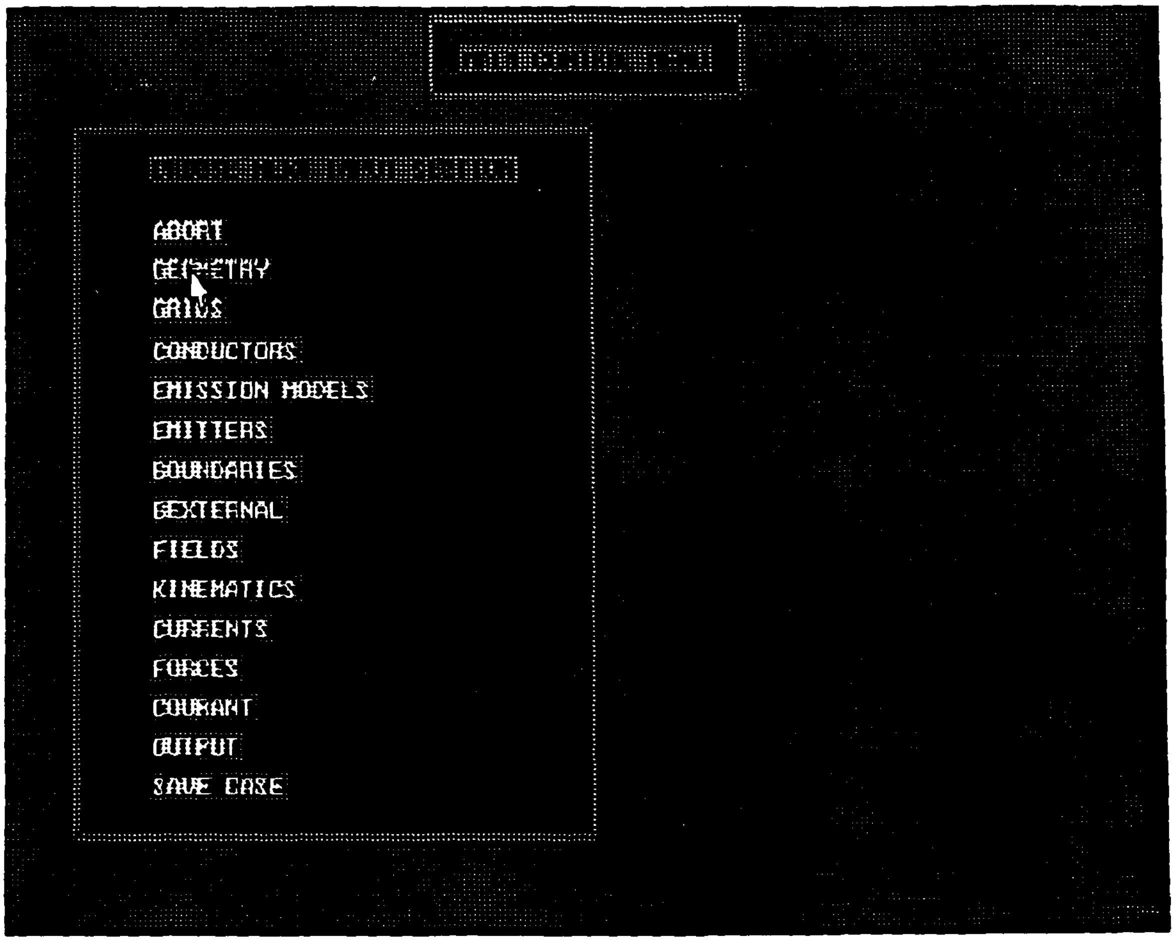

Figure 5 Main Control Menu.

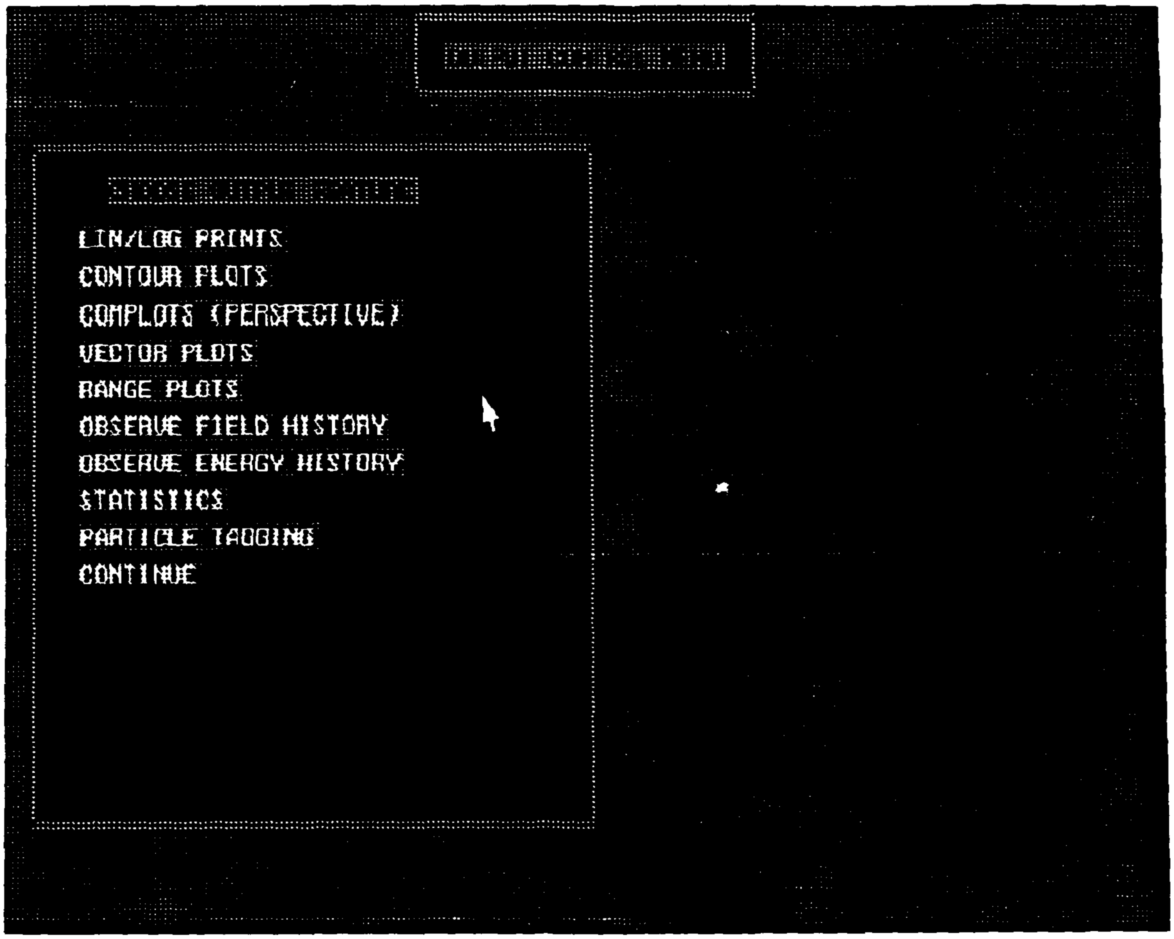

Figure 6 Output Control Menu.

APPENDIX A. THE NATURE OF PARTICLE-IN-CELL SIMULATIONS

Particle-in-cell (PIC) simulation codes [2] are computer programs which simulate the temporal evolution of plasmas, or aggregations of electrically charged particles which interact in a collective manner. Plasmas generally have more complex behaviors than fluids or gases because interactions between plasma components occur through long range electromagnetic forces rather than short range collisions. Plasma components contribute to gross electric and magnetic fields which, in turn, influence the motion of these components. The net result of this intrinsic nonlinearity is that plasmas possess a greater propensity for collective behavior than do fluids or gases. A broad range of physical phenomena can occur in a plasma. Different types of waves and oscillations occur on different space and time scales. Instabilities can arise, lead to turbulence, and transform various types of plasma potential energy into various types of plasma kinetic energy. Activity on one space-time scale and affect activity on another space-time scale. For example, changes in macroscopic properties such as resistivity can result from the turbulent fluctuation of electric fields on a microscopic space-time scale.

Although the behavior of plasmas can be very complex, their governing physical laws are deceptively simple. Maxwell's equations relate electric and magnetic fields to the charge and current density. The plasma charge and current densities are averages of the positions and velocities of individual components. The motion of plasma components is determined from Newton's equation of motion with the Lorentz driving force. While physically intuitive, the mathematics of this process is complex. The laws constitute a coupled set of nonlinear, integro-differential equations which are completely intractable to general solution. The bulk of the theoretical literature in plasma physics deals with the approximate solution of simplified cases of the basic equations.

The physically intuitive statement of the laws governing plasmas does lead to the ability to "solve" for plasma behavior by simulating it in the following algorithmic fashion:

1) Distribute plasma particles in such external fields as desired.

2) Calculate the charge and current densities associated with these particles.

3) Solve the Maxwell equations for the electric and magnetic fields using these calculated charge and current densities.

4) Use the equation of motion and Lorentz force to calculate the advance of the plasma particles over a short duration of time.

5) Repeat steps (2) through (5).

This algorithm is precisely what a particle-in-cell simulation code does. The solution of Maxwell's partial differential equations (Step 3) is generally accomplished by finite difference or finite element techniques and the spatial region of interest is divided into small segments or cells. The calculation of the charge and current densities is accomplished by averaging over particles contained in a cell, thus giving rise to the term particle-in-cell simulation. Once new values for fields have been obtained the individual particles are advanced in time by simple ordinary differential equation techniques.

In light of the description just given, it would seem that almost anyone could perform a particle-in-cell simulation. In point of fact, substantial complexity arises because of the necessity of approximations. Even the fastest computers cannot follow realistic numbers of plasma particles in realistic geometries. Decisions have to be made as to how many plasma "particles" are necessary to adequately describe a particular phenomenon. Even with reduced numbers of particles, storage and computer speed restrict the ability to deal with realistic, three-dimensional spatial configurations. Symmetry considerations must be utilized to reduce the dimensionality of the computation. Plasma phenomena occurs over a very wide range of time scales. The motion of electrons is very rapid--the collective motion of plasma waves and oscillations is much slower, and the transport of plasma due to Coulomb collisions and interactions with electromagnetic fluctuations is slower still. If the focus of attention is on a "slow" phenomena, then approximations must be made to tune the simulation to resolve that time scale. For example, it may be necessary to artificially reduce the mass ratio of ions and electrons in order to bring their time scales closer together. It may be necessary to completely abandon hope of treating electrons as particles-in-cells and instead model them by some statistical method. Such methods are common and are known as hybrid simulations. There are many problems of spatial and temporal resolution and many approximate methods of dealing with these problems.

Almost every aspect of a PIC simulation has a limited range of validity. The number of cells, or the grid spacing, must be small enough to resolve the desired phenomenon without consuming an inordinate amount of time. A large enough number of particles must be chosen in order that the simulation behave sufficiently like a plasma. However, if too many particles are chosen then there is not enough time to study what is desired. When it is necessary to resolve a phenomena on the ion time scale which, nonetheless, depends on much more rapid effects occurring on the electron time scale, the question arises as to how much the artificial mass ratio may be reduced without producing qualitative errors in the phenomena of interest. It is dealing with judgmental questions of these types that generally require the user of a simulation code to be an experienced plasma physicist.

Despite the difficult nature of these judgments, plasma simulations have been performed for more than two decades and a substantial body of knowledge has accumulated. Many commonly occurring situations are very well understood. It is quite reasonable to expect that nonexperts should be able to make useful application of these powerful research tools. However, these novice users must be guided to ensure that they stay well within the accepted ranges of validity of the particular technique they are attempting to use.

The difficulty of allowing novice users access to plasma PIC simulations has two aspects. The first problem is the logistical one of passing the initial information to the simulation code in a format which allows it to operate. This problem is not trivial. Scientists build simulation codes for their own use with little thought to their external use. Even if they were interested in making their codes broadly useful, they are often so enmeshed in the jargon and practice of their specialty that they would have trouble communicating with outsiders. Furthermore, the actual structure of the input information may be complex and voluminous, yet rigidly specified because of the strict syntactical requirements of FORTRAN I/O. This logistical and syntactical problem is tractable within the framework of modern computing and the NTRFACE system is an examples of how contemporary techniques can be brought to bear on this problem.

The second problem of allowing novice users access to plasma simulation codes is the problem of how to also provide access to the expertise of the code developers. Access without such insight is likely to lead to the operation of the model outside it's range of validity. Future versions of the NTRFACE system will address this problem with rule-based expert system techniques drawn from the filed of artificial intelligence.

APPENDIX B. DEVELOPER'S NOTES

The NTRFACE system was developed by one person, a plasma physicist, working half-time for a period of two years. It's design and software architecture reflect the prejudices and programming limitations of the developer. It has not been tested by others and likely contains numerous bugs, omissions, and inconsistencies. However, the developer is very interested in seeing that the system is both useful and usable. Any comments, bug reports, criticisms and/or suggestions are appreciated and will likely lead to changes in subsequent versions of the system. This project grew out of the developer's interest in applications of artificial intelligence and symbolic computing to problems in science and engineering [5-7].

Flexibility and the ability to readily implement modifications was an important consideration in the development of NTRFACE. Therefore, the object-oriented programming paradigm was followed whenever possible. For example, conductors, emitters, and boundaries conditions within MAGIC can all be considered as objects which are "line segments constrained to the computational grid." An emitter object simply has more attributes (parameters of the emission model) attached to it than a conductor object. This type of object-oriented architecture allows the same code to be generic to a number of different system components.

The original NTRFACE system was developed in COMMON LISP. However, serious problems are currently associated with delivery of a PC based software system implemented in LISP. In particular, extended memory (>640K) is required, and LISP development systems address this memory through features of IBM BIOS (basic input/output system) which are not available on many popular IBM clones. To facilitate delivery, the NTRFACE for MAGIC system was developed in TURBO C. As a consequence, NTRFACE for MAGIC has modest memory requirements (< 200K). However, C is less conducive to artificial intelligence programming than LISP and the rule-based expert system techniques for trapping illegal user input which were implemented in NTRFACE have not been incorporated in NTRFACE for MAGIC.

The result of developing NTRFACE for MAGIC in C was an almost universally deliverable (any PC clone with EGA color graphics and mouse) program that was much more compact in memory use, had much more attractive graphics, and was much, much faster than it's LISP prototype. The internal programming, however, is more complicated and rigid. It is considerably harder to add a new feature to the C version than to the LISP version. In addition, some planned extensions such as the incorporation of a rule-based expert system for verifying input would require extensive code development in C but would be relatively straightforward in LISP. An interesting factor was the relative cost of the software development systems. TURBO C costs less than $100 while the GOLD HILL COMMON LISP development system costs nearly $2000. In the course of this redevelopment process, the developer has become quite fond of C. Nonetheless, LISP is far superior for rapid prototyping, flexibility and capacity for dealing with complexity. A hybrid approach which utilized the strong points of both languages is desirable.

The object-oriented programming incorporated within NTRFACE is presently simulated since true object-oriented languages (with method encapsulation and inheritance) are not currently available within full featured, PC based languages. It is anticipated that both C++ (object-oriented extension of C) and CLOS (Common Lisp Object System) will become commercially available for PCs in the next year. Future versions of NTRFACE and NTRFACE for MAGIC will be reformulated in these more powerful and flexible language systems.

Any comments, suggestions and/or criticisms are welcome and should be directed to the developer.

N. T. Gladd
Berkeley, California
July, 1989

Berkeley Research Associates
P. O. Box 241
Berkeley, CA 94701